

**This application is submitted in the name of Inventor Erik P. Staats, Assignor to Apple Computer, Inc., a California Corporation.**

## SPECIFICATION

### AUTOMATIC ID ALLOCATION FOR AV/C ENTITIES

#### CROSS REFERENCE TO RELATED APPLICATION

[0001] This application is a continuation of co-pending United States Patent Application Serial Number 09/432,872, filed November 2, 1999.

#### BACKGROUND OF THE INVENTION

##### 1. Field of the Invention

[0002] This invention relates to ID allocation techniques. More particularly, this invention relates to methods for allocating identification nomenclature to AV/C entities.

##### 2. The Prior Art

[0003] The IEEE 1394 multimedia bus standard is to be the "convergence bus" bringing together the worlds of the PC and digital consumer electronics. It is

readily becoming the digital interface of choice for consumer digital audio/video applications, providing a simple, low-cost and seamless plug-and-play interconnect for clusters of digital A/V devices, and it is being adopted for PCs and peripherals.

**[0004]** The original specification for 1394, called IEEE 1394-1995, supported data transmission speeds of 100 to 400 Mbits/second. Most consumer electronic devices available on the market have supported either 100 or 100/200 Mbits/second; meaning that plenty of headroom remains in the 1394 specification. However, as more devices are added to a system, and improvements in the quality of the A/V data (i.e., more pixels and more bits per pixel) emerge, a need for greater bandwidth and connectivity flexibility has been indicated.

**[0005]** The 1394a specification (pending approval) offers efficiency improvements, including support for very low power, arbitration acceleration, fast reset and suspend/resume features. However, current methods for allocating ID's to new devices are both manual and crude especially when considered in the context of 'hot swappable' devices.

**[0006]** As indicated in the AV/C Digital Interface Command Set General Specification (hereinafter, the General Specification): an AV unit is the physical instantiation of a consumer electronic device, e.g., a camcorder or a VCR, within a

Serial Bus node; an AV subunit is an instantiation of a virtual entity that can be identified uniquely within an AV unit and offers a set of coherent functions; an AV/C is an Audio/video control; and a plug is a physical or virtual end-point of connection implemented by an AV unit or subunit that may receive or transmit isochronous or other data -- plugs may be Serial Bus plugs, accessible through the PCR's (PCR: is a Plug Control Register, as defined by IEC 61883, Digital Interface for Consumer Electronic Audio/Video Equipment; further, an iPCR: is an input plug PCR, as defined by IEC 61883 and an oPCR: is an output plug PCR, as defined by IEC 61883) they may be external, physical plugs on the AV unit; or they may be internal virtual plugs implemented by the AV subunits.

**[0007]** An AV/C target implementation is made up of multiple entities including AV/C subunits and plugs. Each separate entity has an associated ID number used to specify that entity when an AV/C controller sends a command acting upon that entity.

**[0008]** The implementation of the AV/C target device must ensure that the IDs used for the target entities are unique among all entities of the same type. In addition they must be between 0 and n-1 where n is the number of a particular type of entity. Thus an AV/C subunit and plug may both have an ID of 0, but two AV/C subunits may not both have an ID of 0.

**[0009]** The old methods for implementing AV/C target entities are to statically allocate the IDs for each entity. Thus, when implementing the software for the entities, the number of entities must be known in advance. Updating the implementation to support a new entity requires manual allocation of another ID. In addition, removal of an entity requires manual deallocation of its ID, and if its ID ( $m$ ) is less than  $n-1$  (e.g.,  $0 \leq m < n-1$ ), thus, residing somewhere in the middle of the identification listings, the IDs for the entities between  $m+1$  and  $n-1$  must be manually decremented.

**[0010]** Modularity of software components, and independence of implementation between software components, are elements of good software design. However, the manual allocation of IDs described above prevents total independence between the implementations of the AV/C entities. In addition, the manual allocation prevents an implementation of dynamic AV/C entities as would be needed when components are hot swapped into an AV/C device.

### BRIEF DESCRIPTION OF THE INVENTION

**[0011]** This invention provides a means of automatically and dynamically allocating IDs for AV/C entities. The IDs do not need to be determined during the implementation of the entities. The IDs are determined at run time. This has the benefit of allowing an implementation of dynamic AV/C entities.

**[0012]** This invention provides an AV/C entity allocation service which maintains a list of the currently allocated IDs. This list is initially empty. When an AV/C entity is initialized, it calls the allocation service to allocate an ID which it then uses for the initialized entity. The allocation service allocates an ID by starting with an ID of 0. The service then searches its allocated ID list to see if the current ID has already been allocated. If it finds the ID in the list, it increments its current ID and searches the list again. If it does not find the ID, it adds the current ID to the allocated list and returns the ID to the entity.

#### **BRIEF DESCRIPTION OF THE DRAWING FIGURES**

**[0013]** FIG. 1 is schematic overview of the present invention.

**[0014]** FIG. 2 is a schematic drawing of entity/service interaction of the present invention.

**[0015]** FIG. 3 is a flow diagram of the method form allocating IDs of the present invention.

**[0016]** FIG. 4 is an exemplary embodiment of the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

[0017] Persons of ordinary skill in the art will realize that the following description of the present invention is illustrative only and not in any way limiting. Other embodiments of the invention will readily suggest themselves to such skilled persons having the benefit of this disclosure.

[0018] Generally speaking, units, plugs, and subunits are known as entities. According to the General Specification, each entity must have a unique ID associated with it within its class. Referring now to figure 1, an schematic diagram of an exemplary system 10 is depicted. An AV/C unit 12, such as DV camcorder, is shown including a camera subunit 14 and two tape subunits 16 and 18 therein, as well as four external physical plugs 26. Furthermore, the camera subunit includes four virtual plugs 20, tape subunit 16 includes four virtual plugs 22 and tape subunit 18 includes four virtual plugs 24. In viewing the depicted example, 20 entities are indicated. That is, the AV/C unit is an entity (which would be significant if attached to other units), each subunit is an entity, and each plug, both physical and virtual is an entity. Therefore, there are 20 entities depicted within 7 classes (1 unit class, 2 subunit classes, and 4 plug classes).

[0019] Since each entity must have a unique ID associated with it, the AV/C unit would have an ID0 (not shown since no other unit are depicted in figure 1), camera subunit 14 has ID0 associated with it, tape subunit 16 has an ID0

associated with it, but the second tape subunit 18 is ID1. Each set of plugs within each unit or subunit, likewise includes a unique ID as shown.

**[0020]** To allocate these IDs in an ordered fashion, ID allocator service 28 lies within a memory space, such as an EEPROM. Referring now to figure 2, as can be seen schematically, each entity 30-36 is in operative communication with the ID allocator service 28. The ID allocator service 28 serves the function of dynamically allocating IDs to each sensed entity. That is, once an entity is detected, usually on startup, a call is made to the ID allocator service 28 to assign an ID to the new entity. Likewise, when an entity is removed and another like entity is added, a call is made to the ID allocator service 28 to assign the first available unused ID, which may be that of a previous entity.

**[0021]** To accomplish this task, and referring now to figure 3, an ID allocation system 110 is depicted. The system 110 includes as a first activity 112 staring with a current ID equal to zero. If the ID0 is already allocated to an entity, then the system will look to the next ID as in activities 114 and 116. This process will recur until the next available, unused, ID is located. When the next unused ID is located, the newly found entity is assigned that ID by mapping that entity to that ID in an allocation list as in activities 118 and 120. For example, and referring again to figure 1, when the tape subunit 18 was added, the device was detected and a call was made to the ID allocator service 28. The ID allocator service first

checked to see if ID0 was available in the tape subunit class. The service discovered that ID0 was being used already, so it next checked ID1. As ID1 was available, ID1 was assigned to tape subunit 18. No user intervention was required to assign the ID other than adding the entity and turning the system on.

**[0022]** In use and operation, another exemplary schematic 210 is depicted in figure 4. In this example a settop box (212) will act as a bridge between two video cameras on one side of the bridge and two televisions on the other. Included with the settop box are two USB ports 218 and 220 and two 1394 ports 236 and 246. The televisions 238 and 248 are connected to the 1394 ports 236 and 246 respectively via an appropriate 1394 cable. In this example, the televisions are acting as hosts or servers for potential transmissions of video and audio through the STB 212.

**[0023]** It will be understood that included within the STB 212 will be a USB AV/C subunit software module for detecting USB devices on the USB buses. Once a device is connected to one of the USB ports, the USB software will detect the entity and make a call to the ID allocator service as described above.

**[0024]** In this example, then, the camera 214 is first connected via an appropriate USB cable to port 218. The system is turned on, and the new entity is detected by the USB software which builds an AV/C camera subunit 222 and a



virtual plug 228 to put in operative communication with port 218. Plug 228 is an input plug, whereas plugs 232 and 240 are output plugs, and hence AV/C considers them to be of different classes, and as such separate class IDs are associated therewith. The USB software, thus, makes a call to the ID allocator service 226 which initiates its recursive search for an ID as discussed with respect to figure 3. ID0 is then assigned to AV/C camera subunit 222 and then an ID0 is assigned to virtual plug 228. Then, as the bridge serves but one purpose in this example, the subunit 222 must be put in operative communication with ports 236 and 246 via virtual plug 232 and 240 respectively. The ID allocator thus, assigns the next available ID, which in this case is ID0, to the virtual plug 240 and the next ID to virtual plug 232 or ID1 thereby conforming this portion of the system with the General Specification's requirement of unique ID's for each entity.

[0025]        Thereafter, a second camera 216 is added to the STB 212 at port 220. Another call is made to the ID allocator service 226. The ID allocator service then assigns the next available ID, which is ID1 in this case, to the new subunit 224. Again, three virtual plugs are needed to bridge the camera with the televisions 238 and 248 at ports 236 and 246 respectively. Thus, a first virtual input plug 230 is assigned ID0. Then a first virtual output plug 242 is assigned ID0, while a second virtual output plug 234 is assigned ID1. Without the allocator 226, the second subunit could not be built without manually assigning a new ID. As one can appreciate, such is quite a cumbersome and user unfriendly task.

Furthermore, if, thereafter, camera 214 were unplugged from plug 218, the IDs associated therewith would be removed from the ID allocator list and be available for future use automatically in the present system.

**[0026]** While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.